EL COMMODORE 64 Y LOS NIÑOS



MEYER SOLOMON

EDITORIAL NORAY, S.A.

EL
COMMODORE 64
Y LOS NIÑOS

MEYER SOLOMON

EL COMMODORE 64 Y LOS NIÑOS

EDITORIAL NORAY, S.A.
San Gervasio de Cassolas, 79
08022 Barcelona

Título original: My Commodore 64 and me
Traducción de: Ramón Tapias

Primera edición, 1985
Depósito Legal: B.33838-1985
ISBN: 84-7486-052-0
Número de edición de E.N., 84
Printed in Spain - Impreso en España
Litofisan Jaime Piquet, 7 Barcelona

Índice

INTRODUCCION

Este libro es para niños y principiantes absolutos. No intenta ir más allá del mínimo necesario para comprender los principios de la programación. He intentado explicar las cosas concienzudamente y de manera tan simple como me ha sido posible. Espero haber conseguido evitar la presunción de que tras las primeras tres o cuatro páginas el lector se transformará milagrosamente en un experto.

La principal dificultad con los lenguajes de programación no consiste en la esencia sino en el detalle. Hay que comprender un montón de palabras, aprenderlas y colocarlas adecuadamente. Esto es interesante ya que un buen vocabulario puede producir sentencias maravillosas. Pero una serie de palabras no forma necesariamente un vocabulario. Este libro considera las palabras más importantes que (como en cualquier otro lenguaje) son las más simples. Se trata no solamente de las palabras sino de las ideas que dan forma a la programación.

Las operaciones se pueden desarrollar una tras otra, formando una secuencia.

Las operaciones se pueden repetir. Se pueden repetir un número de veces prefijado. O pueden repetirse hasta que ocurra algo que las detenga. Si nunca ocurre nada, continuarán repitiéndose.

Las operaciones pueden dirigirse en una u otra dirección, de acuerdo a una condición.

Las operaciones actuan en los objetos.

Un programa de ordenador es una serie de operaciones sobre objetos. Se facilitan los objetos a las operaciones. Las operaciones procesan los objetos para producir nuevos objetos. Estos últimos pueden ser procesados por otras operaciones del programa o pueden aparecer como resultado final del mismo.

Una de las ideas que este libro no trata es la de la recursión. No es que dicha idea sea particularmente difícil, sino que se utiliza en programas que son más complejos que cualquiera de los que aparecen en este libro. Una hora diaria con el libro, leyéndolo y operando con él, es un programa de trabajos suficiente para comprender cuan simples y potentes son las ideas de la programación.

Creo que el modo más adecuado de proceder a la lectura de este libro sería por parte de los niños y los padres conjuntamente.

Meyer Solomon



PARECE FAMILIAR

Cuando veas por primera vez tu Commodore 64, es posible que te digas: "¡Parece una máquina de escribir!"

¿Has utilizado alguna vez una? Primero se coloca en ella una hoja de papel en blanco. Luego, pulsas las teclas y las palabras aparecen IMPRESAS en el papel. ¿Debe ser blanca la hoja de papel? No necesariamente. Puedes escribir con tinta roja sobre papel negro, con tinta negra sobre papel blanco, y así sucesivamente. ¿Qué ocurre cuando se utiliza una tinta que sea del mismo color que el papel? Si escribes tu nombre con tinta roja sobre un papel rojo, ¿serías capaz de verlo?

¿Has pensado en lo que sucede cuando pulsas la letra A en una máquina de escribir?



Al pulsar la tecla de la A, provocas un movimiento en una serie de palancas de la máquina, se levanta un brazo que presiona la cinta entintada y se imprime una A en el papel.

Si pulsas la tecla A en el teclado de tu Commodore 64, éste obedece tu orden de que imprima la letra A en la pantalla. Pero, por supuesto, utiliza un método diferente al de la máquina de escribir. El Commodore 64 contiene en su interior pastillas electrónicas, o chips, en vez de palancas y engranajes. Muestra cosas en la pantalla al igual que lo hace un aparato de televisión.

¿Está conectado tu Commodore 64?

Observa el signo rectangular parpadeante. Te indica que el Commodore 64 está preparado para recibir las órdenes e instrucciones que tú teclees.



He terminado, así que espero la siguiente orden.

Escribe tu primer apellido. Ahora, el modo en que se le indica al Commodore 64 que ya has terminado una instrucción es pulsando la tecla RETURN. Por lo tanto, pulsa RETURN.

El Commodore 64 te dará un mensaje de error:

?SYNTAX ERROR (Error de sintaxis)

Es un modo bastante descarado de decirte que no comprende lo que quieres ordenarle.

Realmente, esto no es extraño si piensas que tu Commodore 64 es solamente una máquina. Tu aparato de televisión ofrece unas imágenes y sonidos maravillosos, pero es necesario que lo pongas en marcha. Debes cambiar los canales. Tienes que ajustar los controles de brillo, volumen, constraste, sonido.

Del mismo modo, las instrucciones debes dárselas al Commodore 64 en un lenguaje especial que pueda comprender. Este lenguaje se llama BASIC.

Una de las palabras de dicho lenguaje es PRINT.

Escribe:

```
PRINT"COMMODORE 64"
```

Pulsa RETURN para indicarle al Commodore 64 que has terminado tu instrucción. ¿Qué ves en la pantalla? ¿Qué harías si quisieras poner tu primer apellido en vez de COMMODORE 64?

HACER LAS COSAS UNA VEZ Y HACERLAS MAS TARDE

Cuando estas en casa un fin de semana, puedes decidir súbitamente salir a jugar al futbol. Despues, puedes desear leer un libro. Puedes continuar decidiendo hacer cosas a medida que vayas pensando en ellas, sin tener un plan definido. Pero tambien puedes, en vez de eso, hacer una lista de todas las cosas que quieres hacer, una tras otra. Si lo haces así, podrás mirar la lista más tarde y ver lo que tienes que hacer.

Supongamos que tu lista es ésta:

```
10 JUGAR AL FUTBOL
20 LEER UN LIBRO
30 VER LA TELEVISION
40 COMER
50 JUGAR CON EL COMMODORE 64
60 FIN
```

¿Por qué crees que los números van de diez en diez? Vamos a pensarlo.

Supongamos que quieres darte una ducha despues de jugar al futbol. Entonces podrías escribir:

```
15 TOMAR UNA DUCHA
```

Como puedes ver, la disposición de los números, de diez en diez, nos facilita la colocación de alguna cosa más en la lista de cosas a hacer.

Observa cómo te indicas a tí mismo que termines de hacer las cosas de la lista. Para ello colocas la instrucción de la línea 60.

Ahora, tú le puedes ordenar al Commodore 64 que haga las cosas de una en una:

```
PRINT"COMMODORE 64"
```

No olvides pulsar la tecla RETURN tras el comando.

Pero, en vez de eso, puedes darle una lista de órdenes para que el Commodore 64 las obedezca:

```
10 PRINT"COMMODORE 64"
20 END
```

No olvides pulsar la tecla RETURN tras escribir la línea 10 y, tras escribir la línea 20, pulsa RETURN, escribe RUN y pulsa otra vez la tecla RETURN.

Cuando tú le das al ordenador una lista de instrucciones a seguir, él las recuerda. Pero no las lleva a cabo inmediatamente. Seguirá las instrucciones únicamente cuando tú se lo ordenes.

Esta orden se la das con la palabra RUN.

La lista de instrucciones se llama un PROGRAMA.

Si quieres decirle al ordenador que vas a darle un nuevo programa, escribe NEW.

Por lo tanto, escribe NEW.

Ahora escribe este programa:

```
10 POKE 53281,7
20 PRINT"COMMODORE 64"
30 END
```

Observa que la línea 10 es diferente a la línea 10 tecleada antes.

Ahora escribe RUN.

Para devolverle el color normal a la pantalla pulsa las teclas RUN/STOP y RESTORE.

Ahora escribe RUN otra vez.

NUMEROS Y CARACTERES



Veamos de nuevo el programa.

```
10 POKE 53281,7
20 PRINT"COMMODORE 64"
30 END
```

Mira otra vez la línea 20:

```
20 PRINT"COMMODORE 64"
```

¿Por qué no decimos simplemente COMMODORE 64 sin las "comillas"? Inténtalo si quieres y verás que el Commodore 64 no te entiende.

Esto ocurre porque el Commodore 64 necesita saber si estás hablándole de un carácter o de un número.

Los números se pueden sumar, o restar, o multiplicar, o dividir.

1 + 2 = 3
4 − 2 = 2
4 * 2 = 8 (* para el Commodore 64 significa multiplicar Búscalo en el teclado.)

4 / 2 = 2 (/ para el Commodore 64 significa dividir. Búscalo en el teclado.)

Los caracteres son las cosas que no se tratan como si fueran números.

"COMMODORE 64" es una serie de caracteres. No puedes tratarlo del mismo modo que tratarías un número. No puedes hacer cosas como sumarle o restarle algo.

Las "comillas" le dicen al Commodore 64 que se trata de caracteres.

Pero veamos 1984.

1984 parece un número pero tambien se puede tratar como si fuera una serie de caracteres. Depende de lo que quieras significar cuando dices 1984.

Si quieres referirte al año 1984, es una serie de caracteres. Si te refieres al dinero que tienes en el banco, entonces es un número.

Tu nombre es una serie de caracteres. No puedes multiplicar tu primer apellido por tu nombre para obtener un nuevo nombre. Pero sí puedes multiplicar diez por dos para obtener veinte.

¿QUE PASA EN EL INTERIOR?



Esto, una vez más, es la lista de instrucciones de todo lo que puedes hacer en un sábado:

```
10 JUGAR AL FUTBOL
15 DARTE UNA DUCHA
20 LEER UN LIBRO
30 VER LA TELEVISION
40 COMER
50 JUGAR CON EL COMMODORE 64
60 FIN
```

Escribes estas instrucciones en un pedazo de papel. ¡No puedes escribirlas en el aire! Necesitas algo –un pedazo de papel– donde escribir las instrucciones. Incluso, si escribes más instrucciones, necesitarás más papel.

Del mismo modo, cuando introduces un programa en el Commodore 64, se conserva en el espacio que él utiliza para almacenar las instrucciones que tú le das en el programa. Cuanto más largo sea tu programa, más espacio necesitará.

Este espacio se encuentra en la memoria del ordenador. Puedes pensar en la memoria como si estuviera formada por pequeñas cajitas.

¿Significa esto que si mirases en el interior de estas cajitas podrías ver el mismo tipo de letras y números que escribes en un pedazo de papel?

No, nada de eso.

Toma dos pedacitos de papel en blanco. En el primero escribe un 1 en un lado y un 0 en el otro lado. Haz exactamente lo mismo con el otro pedazo de papel.

Ahora coloca los dos pedazos uno junto a otro. Si lo haces así y le das la vuelta a uno de los pedazos y, luego, al otro, obtendrás las combinaciones siguientes:

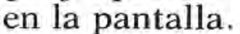
11
10
01
00

Si te imaginas que tú eres el Commodore 64 y que cada una de las combinaciones significa algo para tí, tendrás una idea de cómo almacena el Commodore 64 la información en su memoria.

Por ejemplo, puedes decir que la combinación 11 significa la letra P.

Cada vez que pulsas una tecla, el Commodore 64 cambia las letras en combinaciones como las anteriores. Por supuesto, antes teníamos únicamente cuatro combinaciones. Esto es porque utilizabamos solo dos pedazos de papel. Pero si coges 8 pedazos de papel en blanco y haces exactamente lo que hemos hecho antes, obtendrás 256 combinaciones diferentes. Es más que suficiente para que el Commodore 64 cambie lo que tú escribes en combinaciones de 0 y 1 que pueda entender.

Desde luego, tú no puedes entender el lenguaje que él utiliza en su interior. Por eso, el Commodore 64, muy amablemente, lo cambia al lenguaje que tú entiendes cuando te muestra algo en la pantalla.





ALGUNAS COSAS PERMANECEN IGUALES Y OTRAS COSAS CAMBIAN

Tú naciste con una nariz. Tendrás una nariz toda tu vida (Espero). Cuando algo no cambia, se llama CONSTANTE. ¿Podrías indicar otras cosas a las que puedas llamar constantes?

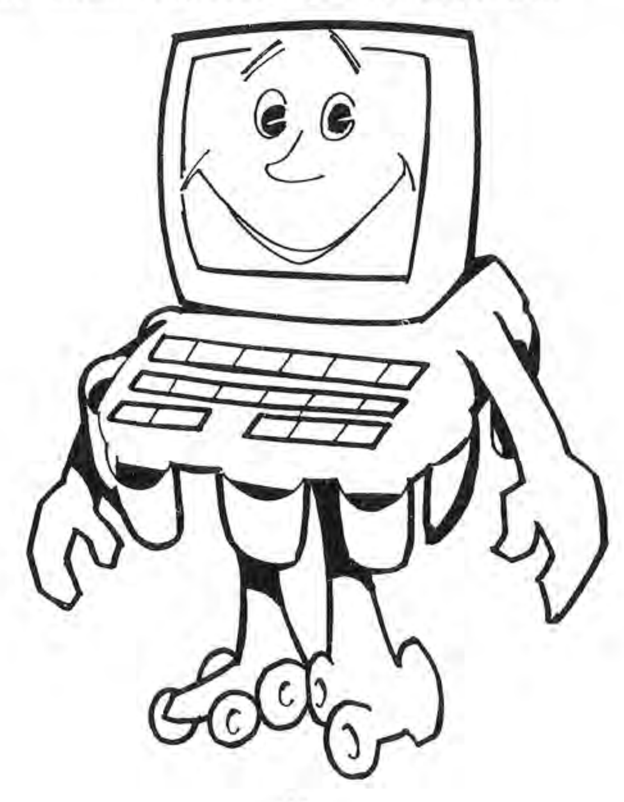
Mira ahora mismo lo que hay en alguno de tus bolsillos. Quizá esté vacío. O quizá haya algo en él, puede que dinero, o un número de teléfono, o el nombre de tu estrella musical favorita. El contenido de tu bolsillo puede cambiar. Las cosas que pueden cambiar se llaman VARIABLES.



Si tu bolsillo contiene cosas que se pueden sumar, restar, multiplicar o dividir, contiene variables NUMERICAS. Por ejemplo, si en tu bolsillo hay cinco pesetas, puedes añadirles dos pesetas para obtener siete. O puedes poner en el bolsillo 15 pesetas, es decir tres veces cinco.

Si tu bolsillo contiene cosas modificables como números de teléfono, nombres o direcciones, contiene lo que el Commodore 64 llama variables de CADENA. Recuerda que no las puedes tratar como si fueran números. No puedes restar tu dirección de la dirección de cualquier otro para obtener una nueva dirección.

Puedes imaginar que el Commodore 64 tiene muchos, muchos bolsillos en la memoria. En estos bolsillos de la memoria conserva las variables numéricas o las variables de cadena. Por supuesto, tambien puede contener, en esos bolsillos de memoria, cosas que no cambian.



UN PROGRAMA MAYOR

Ahora, escribamos el mismo programa que utilizamos antes. Primero, escribe NEW y luego:

```
10 PRINTTT
```

Me he equivocado, pero me he dado cuenta antes de pulsar RETURN. Tengo que corregirlo, claro, y para ello, lo único que he tenido que hacer es pulsar la tecla DELETE para borrar las dos últimas letras T.

A continuación, escribo:

```
10 PRINT A "COMMODORE 64"
```

¡Oh, no! Yo no quería que estuviera ahí esa A. Si la dejo, el Commodore 64 pensará que he es crito tonterías. Por lo tanto, tengo que corregirlo. Primero, tengo que colocar el cursor al lado de la letra A. Esto se hace pulsando las teclas SHIFT y la de flecha a la izquierda/derecha (↔) conjuntamente hasta encontrarme a la derecha de la letra A. Entonces pulso la tecla DELETE hasta que elimino la A. A continuación, pulso la tecla RETURN hasta que el cursor pase a la línea en blanco que está debajo de "READY". Recuerda hacer esto cada vez que hagas una corrección.

Ahora pulso las teclas SHIFT y CLR HOME conjuntamente y escribo LIST. Esto ha sido pesado, pero todos cometemos errores y es por eso por lo que es útil ser capaz de corregir las líneas del programa en vez de volverlas a escribir de nuevo completamente. Esto sería todavía más pesado.

```
10 PRINT"COMMODORE 64"
20 END
```

Ahora vamos a cambiar un poco este pequeño programa y lo vamos a hacer más grande.

Primero, queremos eliminar la línea 10. Lo único que hay que hacer es escribir:

```
10
```

Ahora pulsa RETURN. Este es el modo de eliminar una línea completa de programa.

Queremos decirle al Commodore 64 que reserve uno de sus bolsillos de memoria para una variable NUMERICA. Ahora todos los bolsillos son iguales. De modo que ¿cómo podemos decirle si un bolsillo debe contener un número o una variable de cadena?

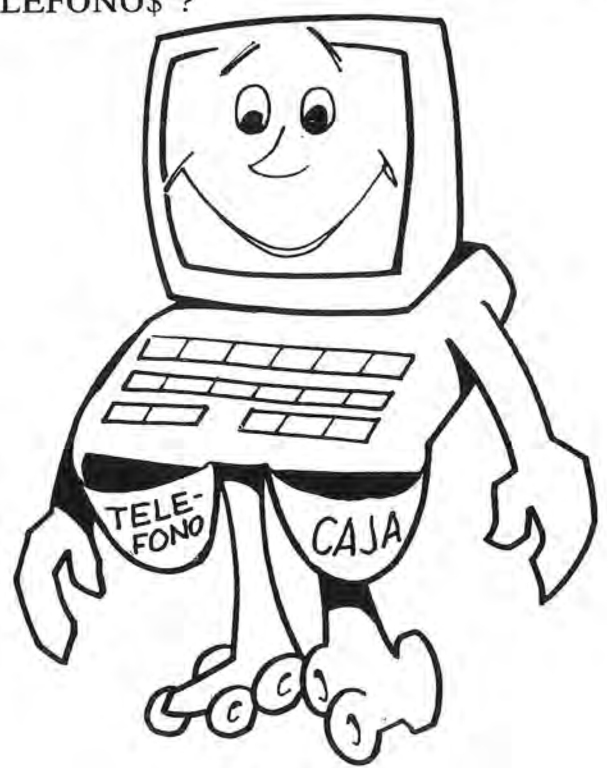
Supongamos que quieres reservar uno de tus bolsillos únicamente para dinero. Un modo de hacerlo (aunque no muy bueno) consistiría en coger una etiqueta adhesiva, escribir en ella "CAJA", y pegarla en el bolsillo.



Si quieres reservar un bolsillo para poner únicamente nombres, podrías coger una etiqueta adhesiva, escribir en ella "NOMBRE$" y pegarla en dicho bolsillo. El signo $ te indica que no estás tratando con números sino con caracteres. Cuando almacenas nombres estás tratando con variables de CADENA. No se puede utilizar la aritmética con los nombres.

¿Qué harías si quisieras reservar un bolsillo para guardar números de teléfono? ¿Un número de teléfono es un número que la gente puede sumar con otro número de telefono? ¿La etiqueta adhesiva correcta para dicho bolsillo sería "TELEFONO$"?





VARIABLES NUMERICAS

Supongamos que tu bolsillo de CAJA empieza sin nada de dinero. Tu CAJA es 0.
Podemos decir:

CAJA = 0



Ahora, supongamos que añades 5 pesetas a lo que hay en el bolsillo de CAJA.

Podríamos decir:

```
CAJA = CAJA + 5
```

¿Cuánto hay ahora en tu bolsillo? Si CAJA empezó siendo 2 y le añadiste 7, ¿cuál es el valor de CAJA que habrá ahora en el bolsillo?

¿Qué harías si quisieras reservar un bolsillo para CANICAS? Escribe una nueva línea 10:

```
10 CAJA = 0
```

Ahora le has ordenado a tu Commodore 64 que reserve uno de sus bolsillos de memoria para una variable NUMERICA llamada CAJA. En el interior de este bolsillo de memoria, él pone el valor 0.

¿Cómo podemos saber lo que el Commodore 64 tiene en el bolsillo de memoria en donde se encuentra la variable CAJA? Al fin y al cabo, como se trata de una variable numérica, podemos modificarla sumándole, restándole, multiplicándola o dividiéndola. No queremos preocuparnos de recordar qué es lo que hay en el bolsillo de la variable, especialmente si lo que hay en él está cambiando continuamente. Preferiríamos que el Commodore 64 hiciese amablemente el trabajo por nosotros y nos dijera lo que queremos saber.

Es fácil. Lo único que tenemos que hacer es ordenarle al Commodore 64 que imprima (PRINT) en la pantalla lo que hay en CAJA.

Para ello, pongamos en la línea 20:

```
20 PRINT CAJA
```

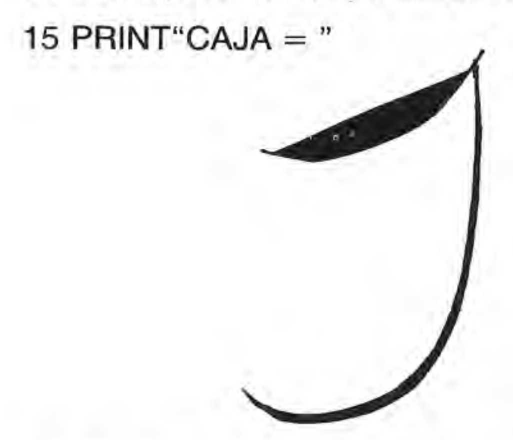
Lanza el programa. No te sorprendas de que

aparezca un cero en la pantalla. Despues de todo, así es como ha empezado CAJA.

Supongamos que un amigo tuyo acaba de llegar y ha visto el 0 y te pregunta qué es lo que significa. Tú podrías decir: "Es la CAJA". ¿Pero por qué malgastar tu aliento si el Commodore 64 puede dar el mensaje en tu lugar?

Para ello, escribe en la línea 15:

```
15 PRINT"CAJA = "
```



Recuerda que si la pantalla se llena de cosas, puedes limpiarla pulsando las teclas SHIFT y CLR HOME conjuntamente. Lista el programa.

Helo aquí:

```
10 CAJA = 0
15 PRINT"CAJA = "
20 PRINT CAJA
```

Ahora lánzalo. Puedes ver como el Commodore 64 imprime el mensaje en la pantalla.

Ahora, digamos que queremos incrementar el contenido del bolsillo de memoria CAJA en un valor de 5.

Escribamos en una línea nueva, la Línea 30:



```
30 CAJA = CAJA + 5
```

Digámosle al Commodore 64 que imprima un mensaje tras esto. Escribe la línea 40:

```
40 PRINT"CAJA = "
```

Ordenémosle tambien al Commodore 64 que nos diga cómo ha cambiado la variable CAJA tras añadirle 5. Escribe:

```
50 PRINT CAJA
```

Limpia la pantalla y lista el programa:

```
10 CAJA = 0
15 PRINT"CAJA = "
20 PRINT CAJA
30 CAJA = CAJA + 5
40 PRINT"CAJA = "
50 PRINT CAJA
```

¿Qué pone el Commodore 64 en el bolsillo de la variable numérica CAJA cuando llega a la línea 10?

¿Qué pone el Commodore 64 en el bolsillo de la variable numérica CAJA cuando llega a la línea 30?

Ahora lanza el programa. No olvides que cada vez que quieras ver las líneas del programa, una vez ejecutado, puedes utilizar otra vez LIST.

Volvamos a tu propio bolsillo. Supongamos que empiezas, una vez más, con el bolsillo vacío. Ahora, supongamos que le añades dos pesetas, tres veces.

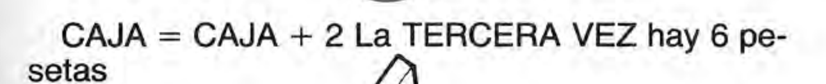
CAJA = 0 Al principio no hay nada



CAJA = CAJA + 2 La PRIMERA VEZ hay 2 pesetas



CAJA = CAJA + 2 La SEGUNDA VEZ hay 4 pesetas



CAJA = CAJA + 2 La TERCERA VEZ hay 6 pesetas

Ahora, supongamos que quieres ser capaz de decirte a tí mismo que sumes a lo que hay en tu bolsillo, sin necesidad de recordar cuántas veces quieres hacerlo.

Supongamos que coges un pedazo de papel y escribes en él estas instrucciones:

```
FOR K = 1 TO 3
CAJA = CAJA + 2
NEXT K
```

Ahora supongamos que coges una etiqueta adhesiva, escribes en ella K y la pegas en otro bolsillo. Ahora, empieza a leer las instrucciones que escribiste.

La instrucción FOR K = 1 TO 3 significa que, la PRIMERA VEZ, habrá un 1 en el bolsillo K. Luego añades 2 pesetas al bolsillo CAJA. Al encontrarte con la instrucción NEXT K, sabes que el bolsillo K contiene un 2, la SEGUNDA VEZ que tienes que añadir 2 pesetas al bolsillo CAJA. Llegamos de nuevo a NEXT K, y el bolsillo K contiene 3, la TERCERA VEZ que tienes que añadir 2 pesetas al bolsillo CAJA.

Tan pronto como K llega a ser superior a 3, tú paras de añadir 2 pesetas al bolsillo CAJA. Esto es así porque la instrucción FOR K = 1 TO 3 te dice que lleves a cabo la operación solamente tres veces.

Intentémoslo de nuevo con números diferentes.

Supongamos que quieres añadir 3 pesetas cada vez a tu bolsillo CAJA. Supongamos que quieres hacerlo 4 veces.

Estas son las instrucciones que deberías escribirte para ello:

```
CAJA = 0
FOR K = 1 TO 4
CAJA = CAJA + 3
NEXT K
```

¿Cuántas veces añades 3 pesetas al bolsillo CAJA? ¿Cuántas pesetas habrá en el bolsillo CAJA tras haberle añadido 3 pesetas durante 4 veces?

Recuerda nuestro programa anterior:

```
10 CAJA = 0
15 PRINT"CAJA = "
20 PRINT CAJA
30 CAJA = CAJA + 5
40 PRINT"CAJA = "
50 PRINT CAJA
```

En la línea 30, el Commodore 64 añade 5 al bolsillo de la variable numérica CAJA. Ordenémosle que lo haga 20 veces.

Escribe la línea 25:

```
25 FOR K = 1 TO 20
```

Ahora escribe la línea 35:

```
35 NEXT K
```

Ahora limpia la pantalla. Recuerda, para ello debes pulsar a la vez las teclas SHIFT y CLR HOME. A continuación, lista el programa. Ahora tenemos:

```
10 CAJA = 0
15 PRINT"CAJA = "
20 PRINT CAJA
25 FOR K = 1 TO 20
30 CAJA = CAJA + 5
35 NEXT K
40 PRINT"CAJA = "
50 PRINT CAJA
```

El Commodore 64 hará todo lo que se encuentra entre las líneas 25 y 35 tantas veces como le indique la línea 25. Esta línea le dice que K debe ir subiendo hasta llegar a 20. Puedes cambiarlo para que K llegue hasta 10 o 30 o cualquier número que quieras. Pruébalo poniendo que K llegue hasta 1000.

Supongamos que quieres que el Commodore 64 le sume 7 al bolsillo de la variable numérica CAJA 40 veces. ¿Que debería cambiar en la línea 25? Pruébalo y recuerda que en la línea 30 le has de decir que le sume 7 al bolsillo CAJA.

SALVALO

¿Recuerdas la lista de cosas que podíamos hacer en un fin de semana? Pues mira, hemos decidido romperla.



Podríamos tirarla o rasgarla y, de ese modo, destruirla.

Supongamos que queremos utilizarla de nuevo algún otro fin de semana. Nos gustaría conservarla y almacenarla en algún lugar seguro. Sería una buena paliza si tuviésemos que escribir las instrucciones de la lista una y otra vez. Por eso salvamos la lista.

Nuestro Programa de Dinero para el Commodore 64 es algo que podríamos salvar antes de desconectar el ordenador. Recuerda, aunque algunas veces tratemos a esta hermosa maquinita como si fuese humana, lo hacemos únicamente porque tenemos imaginación. El Commodore 64 es solamente una máquina y, tan pronto como la desconectamos, se va a dormir y olvida todo lo que le hemos dicho. Por eso debemos salvar

nuestro Programa de Dinero antes de mandar al pequeño Commodore 64 a dormir.



¡Pero todavía no! Solo una vez más, mira el Programa de Dinero de nuevo. Si todavía no está en el Commodore 64, escríbelo. Cambia la línea 25 por:

```
25 FOR K = 1 TO 1000
```

Ahora lístalo y, luego, lánzalo. Ahora lístalo y cambia el 1000 de la línea 25 otra vez a 20.

Pero esta vez, tras escribir RUN, pulsa rápidamente la tecla RUN STOP. El Commodore 64 abandona la ejecución de las instrucciones del Programa de Dinero.



Observa que el Commodore 64 está realmente intentando servirte. Te indica en qué línea del programa se ha detenido. No está mal para algo que no es más que una colección de chips.

Antes de irnos, recordemos que nuestro pequeño Commodore 64 no puede pensar por sí mismo. Te dice donde se ha parado porque tiene un programa en su interior que ya venía preparado cuando compraste el ordenador. Este programa preparado hace tambien cosas como el permitirte copiar tus programas en cinta.

La tecla RUN STOP te permite detener un programa en ejecución. Pero al programa mismo no le ocurre nada. Puedes lanzarlo de nuevo o listarlo o, incluso, modificarlo. Modificar un programa se dice EDITAR un programa. Ya hemos hecho algunas ediciones. RUN STOP te da la oportunidad de cambiar de opinión cuando estás en plena ejecución de un programa.

¡Un minuto! A veces, la gente que escribe programas no quiere que tú puedas utilizar la tecla RUN STOP del modo que acabamos de indicar. Por eso te encontrarás que han impedido que esta tecla actue como lo hace normalmente. Han desactivado la tecla. Perdona por haberte hecho esperar tanto para decirte cómo salvar tu Programa de Dinero. El modo es éste:

Primero, asegúrate de que tu grabador de cassette está conectado tanto al ordenador como a la corriente eléctrica. Ten mucho cuidado con la electricidad. Realmente, es mejor que le pidas a tus padres o a alguna persona mayor que te haga las conexiones.

Pon una cinta –una de las de 15 minutos– en el grabador. Asegúrate de que la cinta está al prin-

cipio y, luego, hazla avanzar un poco de manera que realmente haya cinta magnética en el punto de grabación de tu programa y no un pedazo de plástico inservible.

Ahora escribe:

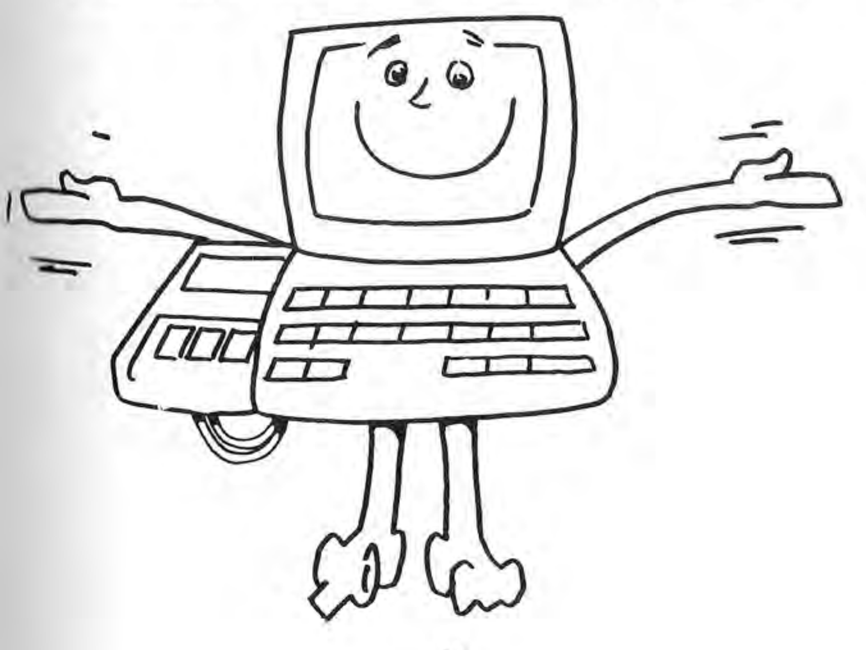
SAVE"DINERO"

Recuerda, nuestro pequeño Commodore 64 no sabe que has terminado de escribir una orden hasta que no pulsas la tecla RETURN. Por lo tanto, pulsa RETURN.

Mira el mensaje que el Commodore 64 te da en la pantalla:

START TAPE AND PRESS ANY KEY

Esto significa: "Pulsa Record y Play en la Cinta". Haz exactamente lo que dice. Si no lo sabes todavía, pídele a alguien que te explique qué bo-



tones debes pulsar para grabar algo en un grabador de cassettes.

Mientras el Commodore 64 esté grabando tu programa DINERO, te indicará que lo está haciendo mostrándote la pantalla en blanco. Si la grabación tiene éxito, verás la palabra READY en la pantalla, una vez vuelva ésta a su estado normal.

Algunas veces, es posible que tu programa no se grabe correctamente. Cuando esto ocurra, pídele ayuda a alguna persona mayor.

Veamos otra vez la instrucción

```
SAVE"DINERO"
```

DINERO es el nombre que le hemos dado a nuestro programa. Pero podríamos darle cualquier otro nombre que quisiéramos, mientras no sea superior a 10 caracteres de longitud. En vez de DINERO, podríamos haber escrito SAVE"ALMAS". Pero es mejor darles a los programas un nombre que te recuerde para qué sirven.

Ahora que hemos aprendido cómo salvar un programa, querremos saber cómo volverlo a introducir en el Commodore 64 desde la cinta en que se encuentra. Por supuesto, debes tener una idea del lugar en la cinta en donde grabaste el programa. Si tienes un cuentavueltas en el grabador, puedes apuntarte en que número empieza y termina el programa.

Supongamos que un programa empieza en el 150 de tu cuentavueltas. Puedes hacer avanzar o retroceder la cinta hasta llegar al 150. A continuación, puedes empezar a cargar tu programa.

1. Acércate todo lo que puedas al principio de

tu programa. Para ello, es posible que tengas que hacer avanzar la cinta.

2. Escribe:

```
LOAD"DINERO"
```

3. A continuación, pulsa RETURN.
4. Cuando veas el mensaje

```
PRESS PLAY ON TAPE
```

en la pantalla, aprieta el botón PLAY.

Verás que la pantalla se queda en blanco y, luego, aparecerá el mensaje "FOUND DINERO" seguido del mensaje de carga. Al poco rato, cuando veas el mensaje:

```
READY
```

querrá decir que el programa está de nuevo en la memoria del Commodore 64.

Si quieres, puedes listar el programa y modificarlo. En otras palabras, EDITARLO. O, por el contrario, puedes decirle al Commodore 64 que siga las instrucciones del programa, utilizando RUN.

Para terminar. Hemos salvado el programa DINERO. Podemos cargarlo más tarde en cualquier momento que queramos. Hagamos una pequeña pausa.

VISUALIZACION

Espero que hayas salvado en cinta el Programa de Dinero. Si lo has hecho, cárgalo y lístalo. Sea como sea, helo aquí de nuevo:

```
10 CAJA = 0
15 PRINT"CAJA = "
20 PRINT CAJA
25 FOR K = 1 TO 20
30 CAJA = CAJA + 5
35 NEXT K
40 PRINT"CAJA = "
50 PRINT CAJA
```



Lanza el programa.

Veamos cuidadosamente lo que el Commodore 64 representa, o visualiza, en la pantalla. Observa que el mensaje CAJA aparece en una línea de la pantalla y el número en la línea siguiente.

Quizá prefiramos tener el número en la misma línea de pantalla en que aparece CAJA. ¿Cómo podemos hacerlo?

Por supuesto, tendremos que modificar o editar el programa, ya que ahora queremos que haga algo diferente.

Utiliza las teclas de cursor arriba y SHIFT para mover el CURSOR junto al 5 de la línea 15.

Utiliza la tecla con la flecha a la derecha para mover el cursor en esa dirección.

Cuando el cursor haya acabado de pasar el final de la línea 15, pulsa la BARRA ESPACIADORA (la barra larga situada en la parte inferior del teclado) una vez. De este modo tendrás un espacio separando una cosa de otra.

Ahora, lo que queremos hacer es que el Commodore 64 imprima lo que hay en el bolsillo CAJA, justo tras el mensaje, en la misma línea. Para ello, el truco podría ser:

```
15 PRINT"CAJA = " ; CAJA
```

Ahora pulsa el signo ; y pulsa la barra espaciadora para dejar un espacio de separación.

Ahora escribe CAJA. Pulsa RETURN para indicarle al Commodore 64 que has terminado de escribir la línea 15. Ahora sigue pulsando RETURN hasta que el cursor llegue bajo la palabra "READY".

¿Qué pasa con la línea 20? Ya no la necesitamos porque su trabajo lo realiza la línea 15. Para eliminarla, escribe simplemente 20 y pulsa RETURN. Si quieres, limpia la pantalla con SHIFT CLR y escribe LIST.

Ahora lanza el Programa de Dinero y comprueba el resultado.

Unicamente por curiosidad, lista la línea 15 y EDITALA de nuevo, pero utilizando esta vez una coma (,) en vez de ;.

Lanza esta versión del programa y observa cómo aparece el resultado esta vez. Ahora lista la línea 15 y vuélvela a dejar como estaba antes.

¿Podemos hacer lo mismo con las líneas 40 y 50? Por supuesto que sí. El ; o la , le indican al Commodore 64 que imprima lo que viene a continuación en la misma línea de la pantalla. ¿Por qué no modificas la línea 40 y eliminas la 50?

Tu nuevo Programa de Dinero deberá quedar así:

```
10 CAJA = 0
15 PRINT"CAJA = " ; CAJA
25 FOR K = 1 TO 20
30 CAJA = CAJA + 5
35 NEXT K
40 PRINT"CAJA = " ; CAJA
```

BAILANDO CLAQUE



Me pregunto si habrás visto alguna vieja película musical en la que alguien baila sobre una pista de baile dividida en cuadrados igual que un tablero de ajedrez.

Imagina que la pantalla es como esa pista de baile de tablero de ajedrez.

La "pista de baile" de la pantalla esta dividida en 40 casillas yendo desde la izquierda hacia la derecha. Cuando vas de arriba a abajo, la "pista de baile" se divide en 25 casillas. Si quieres, puedes dividir un pedazo grande de papel en casillas. Si lo haces, hazlo de 40 casillas de ancho y 25 casillas de alto. Esto te puede ayudar a imaginar mucho mejor lo que estoy diciendo.

Volvamos al Programa de Dinero e imagina que queremos mover la palabra CAJA a través de las casillas de la pantalla.

La instrucción para mover cosas a través de la pantalla es TAB.

Supongamos que queremos mover CAJA cinco casillas. TAB(5) lo hará.

Puedes imaginar que hay un bailarín llamado CURSOR DE TEXTO que lleva CAJA a través de la pista de baile de la pantalla. El número 5 entre paréntesis le indica al CURSOR DE TEXTO que mueva CAJA cinco casillas.

EDITA la línea 15 del Programa de Dinero:

```
15 PRINT TAB(5)"CAJA = " ; CAJA
```

Antes de seguir, pensemos cómo deberíamos EDITAR la línea 15.

Ahora pulsa las teclas de control de cursor y SHIFT para que el CURSOR se coloque junto al 15.

Luego mueve el cursor a la derecha mediante la tecla de flecha a la derecha hasta rebasar la T de PRINT.

Ahora tenemos que introducir otra cosa en ese lugar. Para ello, pulsa las teclas SHIFT e INST DEL, las dos a la vez, varias veces y escribe TAB(5).

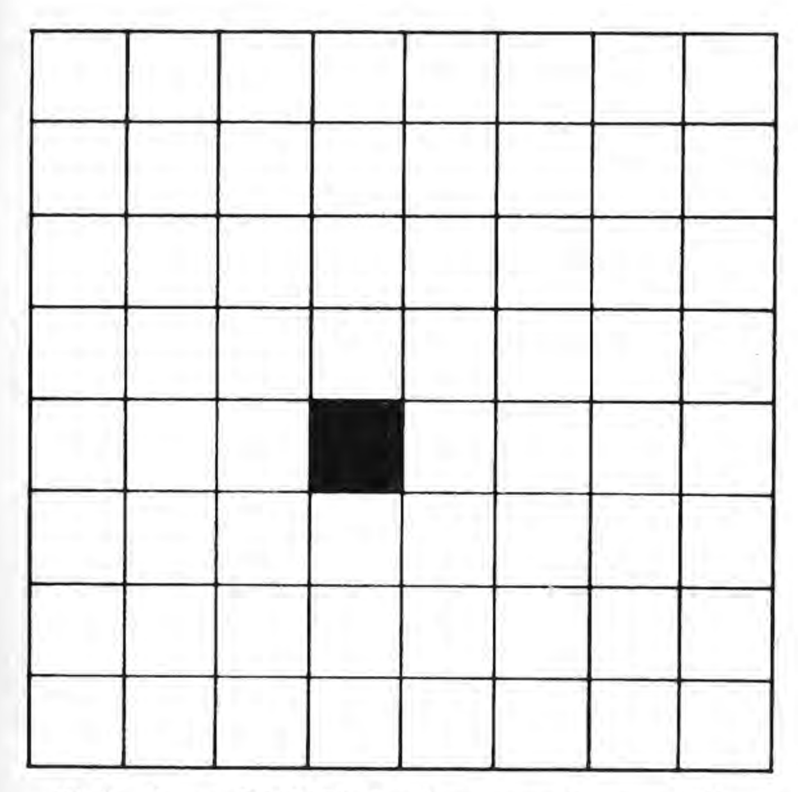
Ahora pulsa RETURN.

(No olvides mover el cursor de nuevo hasta colocarlo debajo de "READY".)

Ahora lista y lanza el programa.

¿Donde empieza su baile el CURSOR DE TEXTO?

Mira la esquina superior izquierda de la pantalla. Imagina que el CURSOR DE TEXTO esta allí, vestido con sombrero de copa, frac y zapatos brillantes.



Si le dices TAB(5), bailará por la pantalla a través de cinco casillas. Una vez hecho esto, soltará lo que lleve.

Ahora modifiquemos la línea 40:

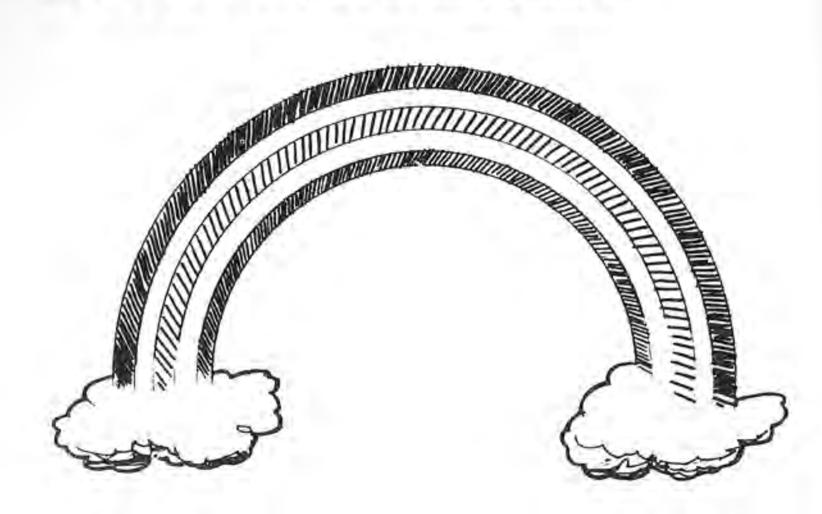
```
40 PRINT TAB(5)"CAJA = " ; CAJA
```

¿Donde crees que bailará el CURSOR DE TEXTO al poner TAB(5)?

Lista y lanza el programa.

Hay una cosa con la que debes llevar cuidado al utilizar TAB. El signo (debe estar pegado a TAB, de otro modo, el Commodore 64 no te va a comprender.

LOS COLORES DEL ARCO IRIS



Hasta ahora has escrito en la pantalla con tinta azul claro sobre papel azul. Al estar la tinta azul claro sobre el papel azul, decimos que el color de arriba es el COLOR DE LOS CARACTERES y el color de abajo es el COLOR DE FONDO.

Recuerda, cualquier cosa que esté encima se dice que es relativa a los CARACTERES. Cualquier cosa que esté debajo se dice que es relativa al FONDO.

Si queremos escribir con tinta verde, podemos decírselo al Commodore 64 escribiendo PRINT"(CTRL) + (6)".

Esto significa, en la línea de programa que escribas: PRINT, abrir comillas ", luego pulsar las teclas CTRL y el número 6 a la vez. En la pantalla aparecerá un símbolo gráfico. Luego cierra las comillas ". Utiliza este método para producir todos los colores de CARACTERES. (Mira la lista que viene a continuación).

Ahora, ¿qué números deberemos utilizar para obtener los colores del papel (el FONDO)? Utilizamos números distintos (mira la lista indicada más abajo) y hacemos "POKE" introduciéndolos en un número de posición de memoria, por ejemplo, escribiendo POKE 53281,7 y pulsando RETURN. El color del fondo cambiará a amarillo.

Hay dieciseis colores de caracteres disponibles en el Commodore 64. Se puede acceder a ellos pulsando la tecla CTRL y un número o la tecla Commodore (C =) y un número tal como se indica a continuación:

	CTRL	tecla			tecla COMMODORE (C >)		
+	1	=	NEGRO	+	1	=	NARANJA
+	2	=	BLANCO	+	2	=	MARRON
+	3	=	ROJO	+	3	=	ROJO CLARO
+	4	=	CIAN	+	4	=	GRIS 1
+	5	=	PURPURA	+	5	=	GRIS 2
+	6	=	VERDE	+	6	=	VERDE CLARO
+	7	=	AZUL	+	7	=	AZUL CLARO
+	8	=	AMARILLO	+	8	=	GRIS 3

Puedes cambiar los colores de los CARACTERES pulsando las teclas CTRL o COMMODORE y el número del color deseado. (Y a continuación RETURN). Los dieciseis colores del FONDO son:

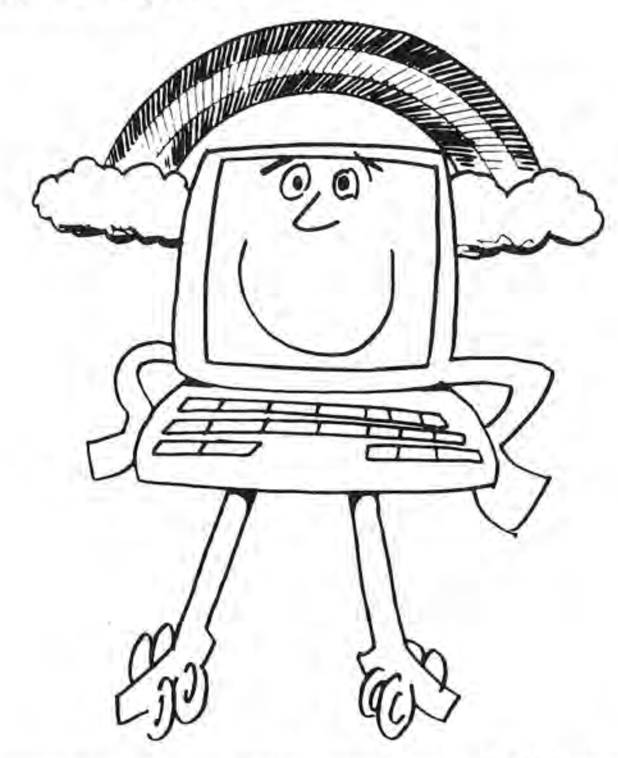
0	=	NEGRO	8	=	NARANJA
1	=	BLANCO	9	=	MARRON
2	=	ROJO	10	=	ROJO CLARO
3	=	CIAN	11	=	GRIS 1
4	=	PURPURA	12	=	GRIS 2
5	=	VERDE	13	=	VERDE CLARO
6	=	AZUL	14	=	AZUL CLARO
7	=	AMARILLO	15	=	GRIS 3

Estos se pueden obtener escribiendo POKE 53281, seguido del número de color deseado (y pulsando RETURN).

Nota: Debes pulsar las teclas RUN/STOP y RESTORE conjuntamente para volver la pantalla al color normal tras modificar los colores del fondo.

Intenta modificar los colores mediante las teclas y utilizando POKE 53281, como hemos dicho antes.

¿Qué es lo que ves?



Supongamos que queremos que CAJA aparezca en tinta VERDE (caracteres) sobre papel AMARILLO (fondo). Escribe una nueva línea, la Línea 12:

```
12 PRINT"(CTRL) + (6)"
```

De este modo obtenemos tinta VERDE (CARACTERES). Escribe otra nueva línea, la Línea 13:

```
13 POKE 53281,7
```

De este modo obtenemos papel AMARILLO (FONDO).

Lista el programa y, luego, lánzalo.

Lista el programa.

Ahora lanza el programa y mira lo que ocurre.

Supongamos que modificamos la línea 13 de modo que ponga POKE 53281,2, para obtener un fondo de papel ROJO. ¿Qué ocurre si escribimos en tinta roja sobre papel rojo? ¿Por qué no lo intentas y miras lo que ocurre al lanzar el programa?

Si lo pruebas, intenta tambien listar el programa.

Si quieres modificar el color del fondo directamente, sin escribir una línea de programa, lo único que debes hacer es escribir el comando POKE 53281, y el número de fondo que quieras. Luego pulsa RETURN.

Así, si ves que no puedes ver el listado porque la tinta y el papel son los dos del mismo color, cambia el color del fondo. Puedes hacerlo directamente escribiendo POKE 53281, y un número distinto al número de la tinta de los caracteres. Prueba con POKE 53281,0. Ahora puedes listar el programa y verlo de nuevo.

Pero, si quieres tener colores de caracteres y de fondo diferentes al ejecutar el programa, debes asegurarte de que las líneas 12 y 13 contengan diferentes colores.

Prueba con distintos colores de fondo y de caracteres.

DULCES 16

Recuerda que el Commodore 64 tiene 16 colores de fondo y 16 colores de caracteres.

Por supuesto, tendrás que EDITAR el programa si deseas probar con nuevos colores.

Si quieres EDITARLO, primero lístalo. Ahora debería estar así:

```
10 CAJA = 0
12 PRINT"(CTRL) + (6)"
13 POKE 53281,0
25 FOR K = 1 TO 20
30 CAJA = CAJA + 5
35 NEXT K
40 PRINT TAB(5)"CAJA = " ; CAJA
```

Ahora pon una cinta en tu grabador de cassettes y salva este programa con otro nombre.

Modifica la línea 13 del modo siguiente:

```
12 PRINT"(CTRL) + (4)"
```

Modifica la línea 13 así:

```
13 POKE 53281,1
```

Ahora lanza el programa.

Puedes pasar un rato distraído utilizando distintos números para el fondo y los caracteres.

DE VUELTA A LOS BOLSILLOS DEL COMMODORE 64



Si lo recuerdas, los bolsillos de la memoria del Commodore 64 puedes contener CADENAS. Para mostrar que lo que hay dentro del bolsillo es una cadena, podemos imaginar que la etiqueta adhesiva del bolsillo lleva escrito el nombre del bolsillo, con un signo $ al final del nombre.

Así que supongamos que tenemos un bolsillo de cadenas que hemos etiquetado A$. Suponga-

mos que queremos poner la cadena “CARA” en su interior.

Ahora vamos a escribir algo nuevo, así que no utilizaremos el Programa de Dinero. Si dicho programa está en el Commodore 64, escribe NEW.

Para crear la línea 10, escribe: PRINT“ luego pulsa las teclas SHIFT y CLR HOME juntas (en la pantalla aparecerá un símbolo gráfico). Escribe “.

```
10 PRINT“(SHIFT) + (CLR HOME)”
20 A$ = “CARA”
```

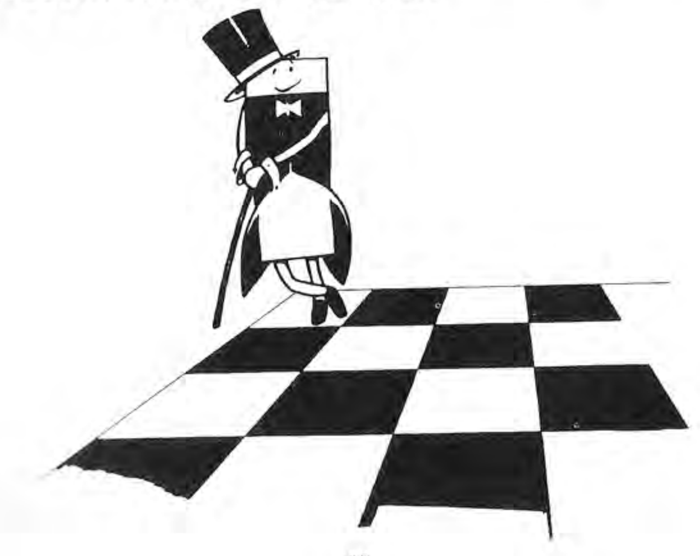
Así colocamos la cadena de caracteres “CARA” en el bolsillo de variables de cadena etiquetado como A$.

Escribe:

```
30 PRINT A$
```

Lanza este pequeño programa.

No olvidemos a nuestro amigo el CURSOR DE TEXTO, el bailarín del TAB.



Lista el programa y modifica la línea 30 de este modo:

```
30 PRINT TAB(5) A$
```

La línea 30 Muestra en la pantalla el contenido del bolsillo de la variable de cadena A$. ¿Qué contiene A$?

Prueba a escribir en tinta roja (caracteres) sobre papel amarillo (fondo).

¿Lo has logrado? He aquí lo que tenemos:

```
10 PRINT"(SHIFT) + (CLR HOME)"
15 PRINT"(CTRL) + (3)"
16 POKE 53281,7
20 A$ = "CARA"
30 PRINT TAB(5) A$
```

Por qué no escribes este programa y lo pruebas?

Cuando yo lo probé, quedaba muy bonito.

Ahora añadamos:

```
35 PRINT TAB(5)"COL"
```

Lancemos el programa.

Bueno, el contenido del bolsillo de la variable A$ no ha cambiado. Todavía tenemos "CARA" en él. Pero prueba a hacer esto:

```
37 A$ = A$ + "COL"
38 PRINT TAB(5) A$
```

Ahora listémoslo y lancémoslo.

¿Qué contiene ahora A$?

Salva el programa del modo habitual. Salvémoslo con el nombre "BICHOS".

No olvides pulsar las teclas RUN/STOP y RESTORE para volver la pantalla a su color normal.

¿Qué significa el signo + en la línea A$ = A$ + "COL"? Lo que significa es que "CARA" y "COL" se unen para formar "CARACOL". A continuación, "CARACOL" es colocado en el bolsillo de la variable de cadena A$. El bolsillo A$, que utilizábamos para contener "CARA". Pero ahora su contenido ha cambiado a "CARACOL".

El poner dos cadenas unidas de modo que se forme una nueva cadena se llama CONCATENACION.

Prueba a escribir directamente, y no en una línea de programa:

```
PRINT"SOL" + "DADO"
```



Ahora escribe NEW y prueba este pequeño programa:

```
10 A$ = "SOL"
20 B$ = "DADO"
30 C$ = A$ + B$
40 PRINT TAB(10) C$
50 END
```

Si miras las líneas 10, 20 y 30, verás que ahora tenemos tres bolsillos de variables de cadena. Estos bolsillos son los llamados A$, B$ y C$.

La línea 10 pone "SOL" en el bolsillo A$.

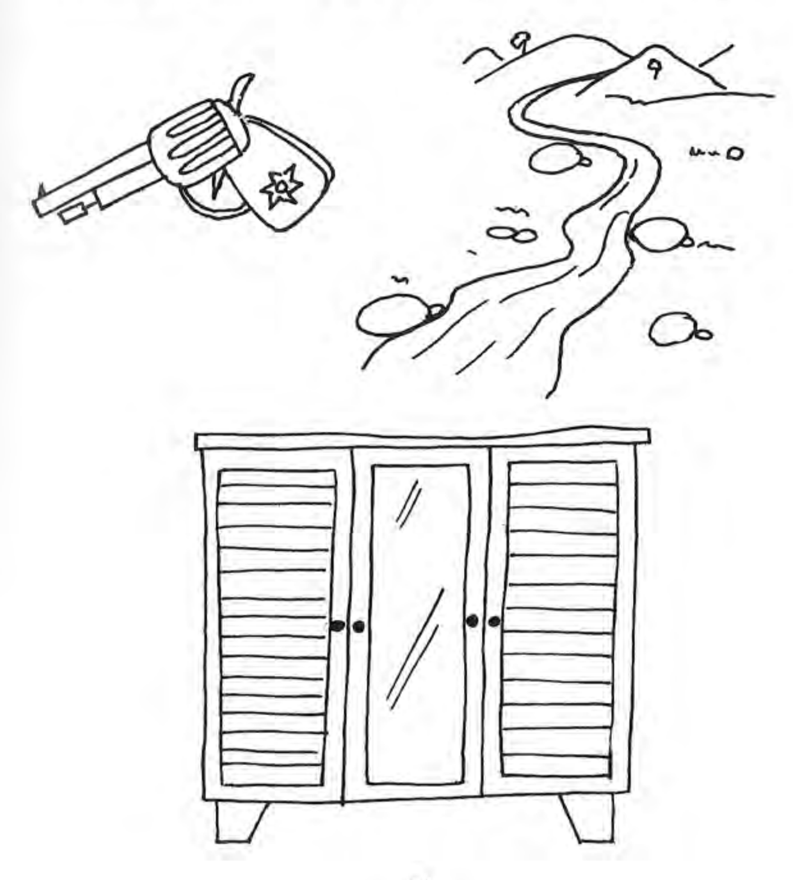
La línea 20 pone "DADO" en el bolsillo B$.

La línea 30 busca lo que hay en ambos bolsillos A$ y B$. Copia "SOL" y "DADO" y lo junta para formar "SOLDADO". Pone "SOLDADO" en el bolsillo C$.

Luego, para terminar, la línea 40 nos muestra, en la pantalla, lo que contiene el bolsillo C$.

CONCATENACION DE CADENAS Y SUMA DE NUMEROS

CONCATENACION es una palabra muy, muy larga. Es un bocado muy grande, pero su significado es muy simple. Simplemente significa poner caracteres uno junto a otro. Así, si pones "ARMA" junto a "RIO", tendrás "ARMARIO"



Recuerda que debes decirle a nuestro pequeño Commodore 64 exactamente lo que debe hacer. Para poner las cadenas una junto a otra tienes que colocar el signo + entre ellas. De otro modo, el Commodore 64 no comprenderá lo que quieres decirle.

Así, para decirle al Commodore 64 que ponga las cadenas "ARMA" y "RIO" una junto a otra, deberás utilizar algo parecido a D$ = "ARMA" + + "RIO". De este modo, el bolsillo D$ contendrá la cadena de caracteres "ARMARIO".

Por supuesto, si quieres que el Commodore 64 muestre lo que hay en el bolsillo D$, tienes que decirle PRINT D$.

Ahora, veamos algo interesante. Escribe:

```
PRINT 3 + 4
```

La respuesta que aparece es 7. Aquí el signo + no coloca el 3 junto al 4 para obtener 34. En vez de eso, nos da 7.

¿Por qué ocurre eso? Bueno, eso es porque no hay comillas " alrededor del 3 y del 4, y de ese modo, el Commodore 64 sabe que se trata de NUMEROS en vez de cadenas. Por lo tanto, SUMA los números 3 y 4 y nos da el resultado de 7.

¡Ajá! ¿Puedes adivinar lo que ocurriría si pusieras las comillas " alrededor del 3 y del 4? ¿Por qué no lo pruebas y lo ves?. Escribe:

```
PRINT"3" + "4"
```

No olvides pulsar RETURN.

CORTANDO LA CADENA

Espero sinceramente que hayas salvado "BICHOS" en cinta. Si lo hiciste, puedes cargar "ROPA" del modo habitual.

Este es el listado:

```
10 PRINT"(SHIFT) + (CLR HOME)"
15 PRINT"(CTRL) + (3)"
16 POKE 53281,7
20 A$ = "CARA"
30 PRINT TAB(5) A$
35 PRINT TAB(5)"COL"
37 A$ = A$ + "COL"
38 PRINT TAB(5) A$
```



Coge un pedazo de papel y escribe en el: CARACOL.

Ahora, empezando por la izquierda, a partir de la letra C, cuenta uno-dos-tres-cuatro hasta que llegues a la letra A. ¿Qué es lo que tienes? Exacto, CARA.

Si cortas el papel justo tras la E, tendrás dos pedazos de papel. El pedazo de la izquierda pondrá CARA.

(De paso, ten cuidado si utilizas tijeras. Pídele permiso a alguna persona mayor antes de utilizarlas. Las cosas que cortan son peligrosas.)

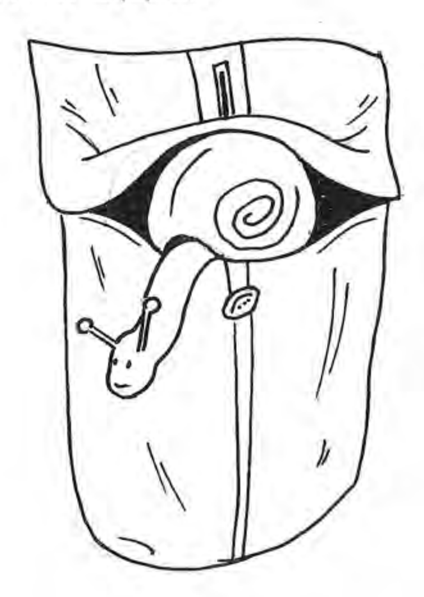
Podemos decirle al Commodore 64 que haga lo mismo con las cadenas de caracteres. Podemos decirle que corte una cadena del mismo modo que hemos cortado CARACOL, empezando por la izquierda.

La palabra especial que lo hace es LEFT$(.

Puedes imaginar que el signo (es como un par de tijeras dispuestas a cortar la cadena.

Añade estas líneas al programa BICHOS:

```
45 L$ = LEFT$(A$,4)
46 PRINT TAB(5) L$
```



Veamos la línea 45. El Commodore 64 busca en el bolsillo A$ y encuentra en su interior "CA-

RACOL". Entonces cuenta cinco caracteres, empezando por el primero de la izquierda, la C. Tras contar los cinco caracteres, tiene la cadena CARA, y la coloca en el bolsillo de la variable de cadena L$.

Luego la línea 46 nos muestra, en la pantalla, lo que contiene L$.

Si hay un LEFT$(, para cortar a partir de la izquierda, debería haber otro comando para cortar a partir de la derecha. Por supuesto que lo hay, RIGHT$(. ¿Qué crees que hará RIGHT$(?

Unicamente por curiosidad, prueba a contar desde la derecha con la palabra ARMARIO.

Escribe ARMARIO en un pedazo de papel. Empieza por la derecha, desde la letra O. Ves hacia atras, contando uno-dos-tres. Has llegado a la letra R. ¿Qué es lo que tienes? Muy bien, RIO.

Si cortas el papel justo antes de la R, tendrás dos pedazos. El pedazo de la derecha pondrá RIO.

Añadamos las líneas 47 y 48 a BICHOS:

```
47 R$ = RIGHT$(A$,3)
48 PRINT TAB(5) R$
```

Veamos la línea 47. El Commodore 64 busca dentro del bolsillo A$ y encuentra la cadena de caracteres "CARACOL". Entonces cuenta tres caracteres hacia atrás, empezando por la L. Obtiene la cadena "COL", y la coloca en el bolsillo de la variable de cadena R$.

La línea 48 nos muestra, en la pantalla, el contenido de R$.

El programa ha quedado bastante cambiado, de manera que ¿por qué no lo salvamos de nuevo? Llámalo BICHOS1 si quieres.

QUE DEBO HACER SI...

¿Recuerdas nuestra lista de cosas para hacer en el fin de semana?

```
10 JUGAR AL FUTBOL
20 LEER UN LIBRO
30 VER LA TELEVISION
40 COMER
50 JUGAR CON EL COMMODORE 64
60 END
```

Añadamos la línea 15:

```
15 TOMAR UN REFRESCO
```

Ahora, cuando leas la lista de cosas a hacer, tendrás un refresco tras el partido de fútbol. Pero, quizá en alguna ocasión no tengas sed. Por lo tanto, solo querrás tomar un refresco cuando realmente tengas sed.

Modifiquemos la línea 15 de este modo:

```
15 SI TENGO SED, ENTONCES TOMARE UN
   REFRESCO
```

Antes, debías tomar siempre un refresco. Ahora, como has utilizado SI y ENTONCES, tomarás un refresco únicamente cuando tengas sed.

SI y ENTONCES (en inglés IF y THEN) son palabras que el Commodore 64 puede comprender.

HOMBRE RICO, HOMBRE POBRE



Imagina que empiezas con 2 pesetas en tu bolsillo de CAJA. Supongamos que una persona muy generosa te da 5 pesetas para que las pongas en tu bolsillo de CAJA. Supongamos que lo hace diez veces. Tan pronto como tu bolsillo de CAJA contenga más de 20 pesetas, tú gritas "SOY RICO! SOY RICO!"

Escribámoslo con cuidado:

MI BOLSILLO DE CAJA CONTIENE 2 PESETAS.

Otro bolsillo contiene 10, para mostrar las veces que la persona generosa me da dinero para añadir a la CAJA. A este bolsillo lo llamaré KONTADOR.

Seguiremos añadiendo 5 a CAJA durante 10 veces, ya que KONTADOR contiene 10.

Si CAJA llega a ser superior a 20, gritaré "SOY RICO! SOY RICO!".

Pero, de cualquier modo, tanto si tengo más de 20 pesetas o no, indicaré cuánto hay en el bolsillo de CAJA.

Por supuesto, tendré que comprobar cuándo he añadido 10 veces el dinero. Ahora probemos a escribir NEW y hacer un programa que el Commodore 64 pueda comprender:

```
10 PRINT"(SHIFT) + (CLR HOME)"
20 CAJA = 2
30 FOR KONTADOR = 1 TO 10
40 CAJA = CAJA + 5
50 IF CAJA > 20 THEN PRINT"SOY RICO!
   SOY RICO!"
60 PRINT"CAJA = " ; CAJA
70 NEXT KONTADOR
80 END
```

Veamos la línea 50. ¿Adivinas lo que significa el signo > en IF CAJA > 20? Correcto, el signo > significa simplemente mayor que.

(Hay otro signo similar, <, colocado al lado del signo mayor que. Significa menor que).

Lanza el programa.

¿Cuánto dinero contiene el bolsillo CAJA al final?

Bien, CAJA termina conteniendo 52.

¿Cual es la utilidad de IF? Es muy útil. Te permite hacer algo únicamente cuando hay un motivo. Te permite tomar una decisión. IF tienes sed, THEN decides tomar un refresco. Solamente llevarás a cabo la línea 15 de tu lista de fin de semana si tienes la razon correcta para hacerlo.

De modo similar, a causa del IF de la línea 50, únicamente gritarás que eres rico cuando tengas más de 20 pesetas en el bolsillo.

Para verlo, cambia la línea 40 de este modo:

```
40 CAJA = CAJA + 1
```

Ahora, el bolsillo de CAJA se incrementa en 1 cada vez, en vez de 5.

Lanza el programa y mira lo que ocurre.

PERO SOLO HASTA...

Supongamos ahora que la persona generosa no quiere ser demasiado generosa.

Supongamos que solo quiere darte 5 pesetas hasta que tu bolsillo de CAJA contenga más de 20 pesetas. Entonces quiere detenerse.

Imagínalo tú mismo. Tienes dos pesetas en el bolsillo de CAJA. El te da 5 pesetas. Tu las añades a las 2 de tu bolsillo de CAJA. ¿Tienes más de 20 pesetas? No. Por lo tanto, te da otras 5 pesetas. Y así sucesivamente. Repetirá la misma operación pero únicamente hasta que tengas más de 20 pesetas.



IF y THEN son dos palabras que el Commodore 64 comprende.

Escribamos un pequeño programa utilizando IF y THEN. Recuerda utilizar NEW para indicarle al Commodore 64 que vas a escribir un nuevo programa.

Escribe el programa siguiente. Verás huecos en el, pero no intentes copiarlos. Simplemente escribe las líneas como siempre, una tras otra. Por supuesto, cuando termines de escribir una línea, debes utilizar RETURN para que el Commodore 64 lo sepa.

```
20 PRINT"(CTRL) + (7)"
30 POKE 53281,1
40 PRINT"SOY POBRE!"
50 CAJA = 2
80 CAJA = CAJA + 5
90 PRINT"CAJA = " ; CAJA
100 IF CAJA < 20 THEN GOTO 80
110 PRINT
120 PRINT
130 PRINT"ERES RICO."
140 PRINT"POR LO TANTO, NO HAY MAS DI-
    NERO PARA TI!"
150 END
```



Mira las líneas 110 y 120 del programa. Estas líneas solo imprimen líneas en blanco. Las líneas en blanco separan el mensaje "ERES RICO" de los otros mensajes de la pantalla. Lanza el programa y mira lo que ocurre. Luego, si quieres, elimina las líneas 110 y 120 y mira lo que ocurre cuando lanzas de nuevo el programa.

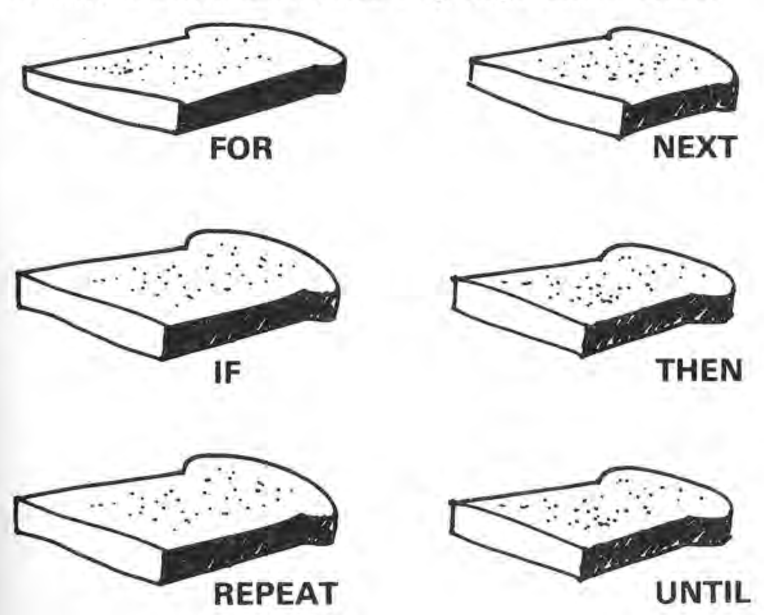
¿Recuerdas cómo se elimina una línea completa? Simplemente escribe el número de línea y pulsa RETURN.

De paso, estoy seguro de que habrás notado lo siguiente:

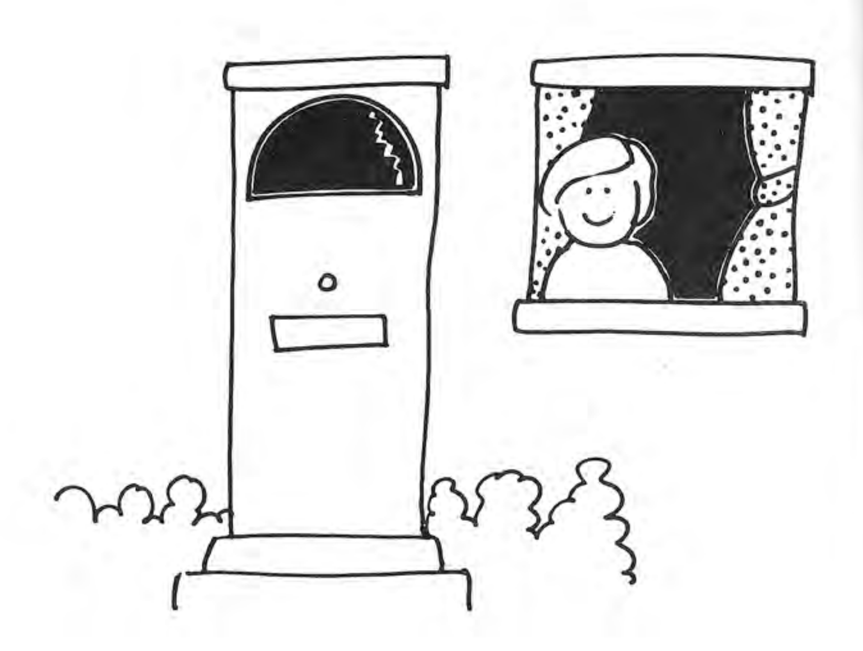
FOR siempre va unido a un NEXT.

IF siempre va unido a un THEN.

Van siempre juntos como las dos rebanadas de un bocadillo. Y, al igual que un bocadillo habitualmente lleva algo entre las dos rebanadas, tambien habitualmente hay algo entre ellos.



EL COMMODORE 64 ESPERA...

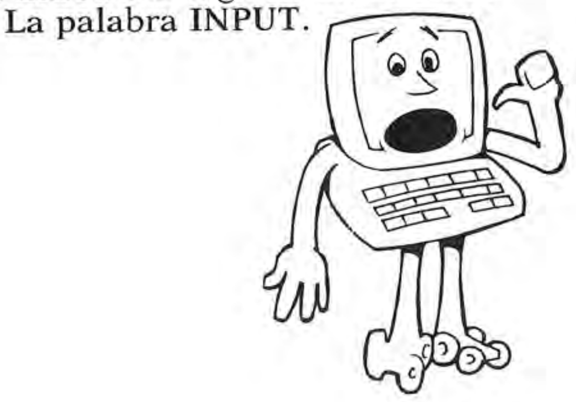


Si estás esperando una carta el día de tu cumpleaños, es posible que te levantes temprano. Quizás te quedes al lado de la puerta, esperando que el cartero introduzca una carta en el buzón.

Del mismo modo, puedes hacer que el Commodore 64 espere a que tú introduzcas algo en uno de los bolsillos de su memoria.

¿Qué palabra necesitará el Commodore 64 para introducir algo en un bolsillo?

La palabra INPUT.



El Commodore 64 puede esperar a que le des un número para colocarlo en un bolsillo de variable numérica. O puede esperar a que le introduzcas una variable de CADENA de caracteres en un bolsillo de variable de cadena.

¿Cómo sabe qué tipo de variable debe introducir en cada bolsillo?

Bueno, una vez más, todo consiste en ver si hay o no un signo $, que es el que le indica al Commodore 64 que debe esperar una variable de cadena de caracteres.

Por ejemplo, si le dices al Commodore 64 que haga INPUT N$, pensará que lo que escribes es una cadena de caracteres. Lo pondrá en el bolsillo de variable de cadena N$.

Pero si le dices al Commodore 64 que haga INPUT N, pensará que lo que escribes es un número. Pondrá ese número en el bolsillo de variable numérica N.

PIENSA UN NUMERO

Ahora escribe NEW y, a continuación, escribe este corto programa:

```
10 PRINT"(SHIFT) + (CLR HOME)"
20 PRINT"(CTRL) + (3)"
30 PRINT TAB(3)"ESCRIBE"
40 PRINT TAB(3)"TU NUMERO"
50 PRINT TAB(3)"FAVORITO"
60 PRINT TAB(3)"DESDE 1 A 9"
70 INPUT NUM
80 PRINT TAB(1)"TU FAVORITO ES";
90 PRINT TAB(80)NUM
100 NUM = NUM * 9
110 RESPUESTA = NUM * 123456789
120 PRINT TAB(3)"AQUI TIENES UN MONTON
     DE";
130 PRINT"(CTRL) + (5)"
140 PRINT TAB(6) RESPUESTA;
150 PRINT"(CTRL) + (3)";
160 PRINT TAB(8)"!!!"
170 END
```

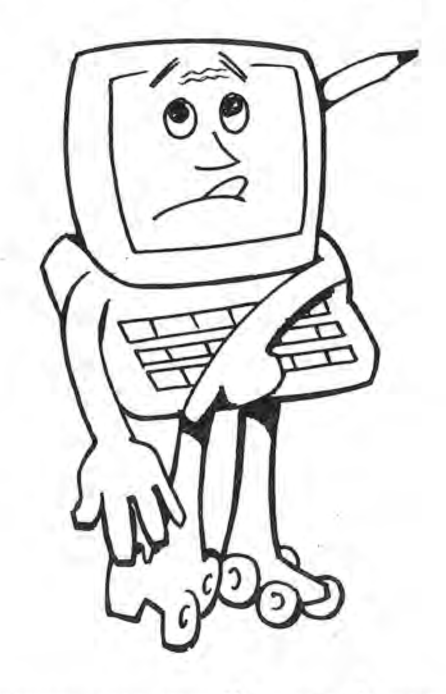
Ejecuta el programa. Ahora lístalo.

Veamos la Línea 70. Si hubieras escrito NUM = 8 en la Línea 70, el número 8 ya habría estado en el bolsillo de la variable numérica NUM. Tendrías que limitarte a ese valor y no podrías elegir un número entre 1 y 9.

Pero, como la Línea 70 lleva la palabra INPUT, el Commodore 64 espera a que tú escribas un número, que es el que pondrá en el bolsillo NUM.

Para indicarte que está esperando que le digas algo, el Commodore 64 te muestra un ? en la pantalla.

Veamos la Línea 100. Aquí, lo que está contenido en el bolsillo NUM se multiplica por 9.



Supongamos que escribes un 4 como número favorito en el momento en que ves el signo ?. El bolsillo NUM contendrá el 4. Tras hacer lo que dice la Línea 100, el bolsillo NUM del Commodore 64 contendrá 36.

Veamos la Línea 110. Aquí, NUM se multiplica por un número muy largo. Pero no tenemos que preocuparnos porque el Commodore 64 hará este trabajo por nosotros. Pondrá el nuevo número en el bolsillo RESPUESTA.

Ya deberías comprender todo lo que hacen las demas líneas del programa. Pero veamos una vez más el programa línea por línea. Mira el listado en la pantalla a medida que lees el parrafo siguiente.

La Línea 10 le dice al Commodore 64 que limpie la pantalla.

La Línea 20 pone el número de color 8 para los CARACTERES.

La Línea 30 utiliza TAB para imprimir parte de un mensaje. TAB baila a través de 3 espacios en la pantalla.

La Línea 40 utiliza TAB para otra porción de mensaje. Baila 3 espacios a través de la pantalla.

La Línea 50 utiliza TAB para otra porción de mensaje. Baila 3 espacios a través de la pantalla.

La Línea 60 utiliza TAB para la última parte del mensaje. Baila 3 espacios a través de la pantalla.

La Línea 70 muestra un signo ? y hace que el Commodore 64 espere a que tú escribas un número. Ese número lo pone en la variable numérica N.

La Línea 80 utiliza TAB para mostrarte otro mensaje en la pantalla.

Utilizamos números diferentes para TAB porque queremos poner cosas en sitios distintos de la pantalla.

La Línea 90 utiliza PRINT y TAB para mostrarnos, en un lado de la pantalla, lo que contiene NUM en este momento.

La Línea 100 multiplica el contenido de NUM por 9. De ese modo NUM pasa a contener 9 veces lo que tenía antes.

La Línea 110 multiplica NUM por un número muy largo. No tenemos que molestarnos en cal-



cularlo porque el Commodore 64 lo hace por nosotros. Ese número lo pone en la variable numérica RESPUESTA.

La Línea 120 coloca otro mensaje en la pantalla, utilizando TAB.

La Línea 130 cambia el número de color de los caracteres a 5. Todo lo que se imprima a continuación saldrá en este nuevo color.

La Línea 140 utiliza PRINT y TAB para mostrar el contenido de RESPUESTA, en un lado de la pantalla.

La Línea 150 modifica el número del color de los CARACTERES a 3.

La Línea 160 imprime únicamente cuatro signos !!!! en un lado de la pantalla, utilizando TAB.

¡QUE NOTABLE!

Hemos visto todas las líneas del programa pero, por supuesto, tendremos que saber mejor las cosas y necesitaremos menos explicaciones.

Si queremos poner algunas pequeñas explicaciones en el propio programa, de manera que lo pueda comprender cualquier otro, podemos utilizar REM. REM simplemente le dice al Commodore 64 que lo que va a continuación, en la línea, es una nota o una observación. Una observación te indica algo sobre una parte del programa. El Commodore 64 no hace nada cuando, durante la ejecución de un programa, se encuentra un REM. La línea REM únicamente aparece cuando listas el programa.

Por ejemplo, podemos añadir la Línea 145, antes de la Línea 150:

```
145 REM COLOR DE LOS CARACTERES
```

Podemos añadir otras líneas REM del mismo modo, si queremos explicar alguna parte importante de un programa. Es muy útil en un programa realmente largo, que podría ser difícil de leer y de comprender sin una explicación de los puntos más importantes.

Pero, recuerda, demasiadas líneas REM tambien pueden hacer que un programa sea difícil de leer. Tienes que utilizarlas con sentido comun.

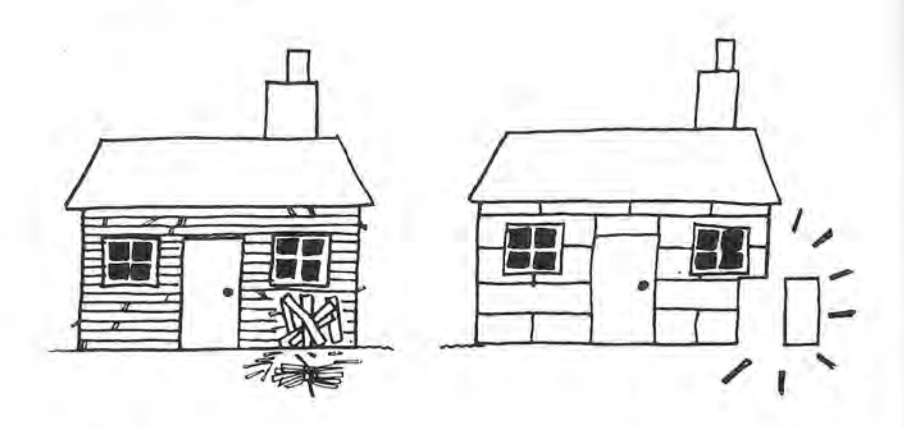
INSPIRA Y SOPLA

Tengo que admitirlo, ha pasado mucho tiempo desde que iba a la escuela. Pero me parece recordar una historia referente a tres cerditos que construyeron una casa cada uno. No puedo recordarla completamente, pero estoy seguro de que uno de los cerditos construyó su casa con paja y otro la construyó con ladrillos. Y había un gran lobo malvado que inspiró y sopló y derribó la casa de paja. Pero quedó completamente agotado de tanto inspirar y soplar cuando intentó derribar la casa de ladrillos.



¿Qué tiene que ver esto con los programas? Tiene mucha relación.

Una casa de paja es sucia. Se cae a pedazos y tienes que estar siempre remendándola. Si hay un pedazo de pared que necesita una reparación, tienes que quitar tambien pedazos de pared en buen estado y la paja que pongas es muy difícil que encaje exactamente en el lugar donde quieras ponerla.



Si tu casa está construida de ladrillos, es fácil quitar un ladrillo estropeado y poner uno nuevo. Es fácil reemplazar una parte por otra.

Del mismo modo, debes escribir tus programas de modo que sean limpios y construirlos a base de pedazos pequeños. Cada pedazo debe tener un propósito claro. Un ladrillo tiene una forma sólida y clara. encaja limpiamente con otros ladrillos.

¿Es más fácil decirlo que hacerlo? Al principio, sí. Pero, con la práctica, llegará a ser tan fácil hacerlo como decirlo.

Quiero escribir un juego.

1. Primero, quiero PREPARAR LA PANTALLA.

2. Luego quiero decirle al Commodore 64 que cree un bolsillo de variable de cadena y un bolsillo de variable numérica.

3. Quiero darle algunos mensajes al jugador para EXPLICARLE COMO FUNCIONA EL JUEGO.

4. Quiero preguntarle al jugador si quiere jugar o no. Tan pronto como quiera dejar de jugar, el juego terminará. De otro modo continuará el juego.

5. Luego quiero presentarle un problema al jugador. Quiero que adivine si el lobo esta "inspirando" o "soplando". De eso es de lo que se trata el juego.

6. Mientras el jugador intenta adivinar lo que está haciendo el lobo, quiero tambien comprobar si lo ha adivinado. ¿Ha GANADO? Si ha ganado, tengo que hacer esto:
Tengo que decirle que ha ganado y, a continuación, terminar el juego.

7. Quiero decirle cuántos intentos lleva hechos. Tambien quiero comprobar si ha hecho más de 3 intentos. Si los ha hecho, el lobo ha inspirado y soplado suficientes veces como para derribar su casa. Por lo tanto, ha PERDIDO. Tengo que darle un mensaje y terminar el juego.
Mientras el jugador no haya hecho más de 3 intentos, puede seguir jugando hasta que quiera pararse.

Si piensas escribir cualquier tipo de programa, verás que utiliza el mismo tipo de ladrillos de programa:

Un ladrillo para mostrar las cosas en la pantalla al comienzo del programa.

Un ladrillo para crear ciertas variables.

Un ladrillo para dar mensajes referentes al programa.

Un ladrillo para introducir la información que el programa utilice.

Un ladrillo para crear alguna salida de información facilitada por el programa. Puede tratarse de un problema para el usuario, como en un juego. Pero en vez de eso puede ser una solución, como cuando los ordenadores suman una serie de números y nos dan el resultado en la pantalla.

Un ladrillo para ver si el programa tiene un resultado igual al del jugador. En ese caso, ¿ha ganado el jugador?

Un ladrillo para ver si el programa tiene un resultado distinto al del jugador. En ese caso, ¿ha perdido el jugador porque ha utilizado todos sus intentos?

Un ladrillo para mantener informado al usuario de lo que está ocurriendo. En este caso, ¿cuántos intentos.

Cada ladrillo de PROGRAMACION se llama una SUBRUTINA. Observa que no hemos hecho ningún programa. Pero hemos hecho algo más importante que escribir líneas de programa. Hemos planificado un programa para construirlo con ladrillos de SUBRUTINAS.

EL JUEGO DE INSPIRA Y SOPLA

1. Describiremos o DEFINIREMOS el ladrillo de la SUBRUTINA DE PREPARACION DE LA PANTALLA. Démosle un nombre para recordarla con el. Llamémosla PREPARACION. Por lo tanto nuestra descripción se puede llamar PREPARACION. Hela aquí:

PREPARACION

El color de los caracteres es rojo.
El color del fondo es amarillo.

RETURN

Observa el RETURN. ¿Qué crees que significa? Correcto, significa que hemos terminado la definición de una subrutina.

2. DEFINAMOS la SUBRUTINA que crea las variables que necesitamos para Inspira y Sopla. Llamémosla VARIABLES.

VARIABLES

Pregunto el nombre del jugador. Es decir, hago un INPUT para poner el nombre en una cadena de caracteres. La pongo en un bolsillo de variable de cadena llamado NOMBRE$.

Le digo al Commodore 64 que cree un bolsillo de variable numérica llamado JUGADAS conteniendo un 0. (JUGADAS se incrementará en 1 cada vez que el jugador haga una jugada.) JUGADAS le dirá al jugador cuántas jugadas lleva hechas hasta el momento.

RETURN

3. DEFINO la SUBRUTINA que le dice al jugador las REGLAS del juego.

REGLAS

Imprimo "Hola" (Y el nombre).
Imprimo "El juego de Inspira y Sopla es bonito y dificil"
Imprimo "¿Inspiro o Soplo?"
Imprimo "escribe INSPIRAS o SOPLAS y pulsa Return"
Imprimo "Suyo afectísimo, el Lobo"

RETURN

Ahora tenemos que continuar repitiendo el problema (lo cual forma el ladrillo de subrutina número 5) HASTA que el jugador escriba "P" para parar el juego. El número 8 es el que lo hace.

Así que "escuchamos" simplemente diciendo GET A$. Vamos comprobando A$. Si es igual a "S", el juego se repetirá al llegar a la línea 880. También le decimos al jugador que pulse P para parar el juego si es lo que desea. Si quiere seguir jugando, le diremos al jugador que pulse S. Tan pronto como pulse la P, se termina el juego.

4. Esta es la subrutina PROBLEMA. La llamaremos así. Tan pronto como el jugador intente resolverlo, incrementaremos JUGADAS en 1.

El problema es: ¿está el Lobo inspirando o soplando? ¿Cómo lo haremos para que el Commodore 64 elija entre hacer que el Lobo inspire o sople?



Hay una palabra muy útil que se llama RND. Si tenemos los números 1, 2, 3, y así sucesivamente, escritos en pedazos de papel separados, y cerramos los ojos y elegimos uno, estamos haciendo una elección AL AZAR. No puedes estar seguro de cual es el número que eliges. RND le dice al Commodore 64 que cierre los ojos y elija un numero.

Si pones INT(RND(1)*12) + 1, elegirá un número entre 1 y 12.

Si pones INT(RND(1)*2) + 1, elegirá 1 o 2. Ya lo tenemos. Vamos a utilizar esto. Si el número que sale es el 1, pondremos "INSPIRAS" en el bolsillo de variable de cadena R$.

Si la función RND elige el 2, pondremos "SOPLAS" en R$.

PROBLEMA

Primero, añado 1 a la variable numérica JUGADAS. Así, JUGADAS = JUGADAS + 1.

Hago una variable numérica igual a INT(RND(1)*2) + 1. A esta variable numérica la

llamo R. Ahora cambio a rojo el color de los caracteres. Si R es 1, imprimo TAB(1) y LOBO. Si R es 2, imprimo TAB(14) y LOBO.

Ahora, si R = 1, digo que R$ = "INSPIRAS". Si R = 2, R$ = "SOPLAS".

Ahora le preguntamos al jugador: "¿estoy inspirando o soplando?".

¿Qué es lo que ha contestado el jugador? Le decimos al Commodore 64 que espere a recibir INSPIRAS o SOPLAS. Pondremos algo parecido a INPUT A$. Si es "INSPIRAS", lo comprobaremos con R$ para ver si son iguales. Si lo son, le diremos al jugador que ha ganado, utilizando la subrutina GANADOR. Si R$ no es igual a A$, pasaremos a decirle al jugador cuántas jugadas lleva. Haremos lo mismo en el caso de que el jugador haya escrito "SOPLAS". Si el jugador escribe algo totalmente diferente, simplemente pierde una tirada.

RETURN

5. La subrutina GANADOR imprime el nombre del jugador, le dice que ha ganado y da un mensaje parecido a "No-puedo derribar tu casa. Suyo frustradisimo, Lobo." Luego le decimos al ordenador que termine.

6. Le decimos al jugador cuántas jugadas lleva. Si son más de 3, hacemos una subrutina que podemos llamar subrutina INSPIRA-SOPLA. En caso contrario, repetimos el problema.

7. INSPIRA-SOPLA puede imprimir mensajes como "es tu intento numero cuatro, tu casa esta hecha de paja. He inspirado y soplado y he derribado tu casa. Suyo contentísimo, Lobo". Luego terminamos el juego.

Si el jugador no ha dicho que quería terminar, o no ha ganado, o no ha utilizado todas sus oportunidades, esto significa que A$ no es "P". Por lo tanto, el juego se repite.

Hemos dejado que seas tú mismo el que construyas las partes 5, 6 y 7 como subrutinas.

¡EMPIEZA CON TUS PROGRAMAS!

Hagamos una lista de las subrutinas, junto con otros pedazos del juego Inspira y Sopla:

```
10 PRINT"(SHIFT) + (CLR HOME)"
20 GOSUB 140
30 GOSUB 190
40 GOSUB 240
100 GOSUB 340
110 GOSUB 550
120 IF JUGADAS > 3 THEN 10
130 GOTO 100
```

Este el el conjunto del juego de Inspira y Sopla, escrito como una lista de subrutinas. Cada vez que el Commodore 64 llega a un GOSUB, salta a la línea indicada y vuelve cuando encuentra el comando RETURN.

Así, cuando el Commodore 64 llega a la LINEA 20 y ve GOSUB 140, busca esto:

```
140 REM PREPARACION
170 PRINT"(SHIFT) + (CLR HOME)"
180 RETURN
```

Así que sigamos con las definiciones de todas las subrutinas.

```
190 REM VARIABLES
200 PRINT"¿CUAL ES TU NOMBRE?"
210 INPUT NOMBRE$
220 JUGADAS = 0
230 RETURN

240 REM REGLAS
250 PRINT"ESTE ES EL JUEGO DE INSPIRA Y
    SOPLA"
260 PRINT"ESCRIBE INSPIRAS SI CREES
    QUE ESTOY INSPIRANDO"
270 PRINT"ESCRIBE SOPLAS SI CREES QUE
    ESTOY SOPLANDO"
280 PRINT"LUEGO PULSA RETURN"
290 PRINT"SUYO AFECTISIMO, LOBO"
330 RETURN

340 REM PROBLEMA
350 JUGADAS = JUGADAS + 1
360 R = INT(RND(1)*2) + 1
370 IF R = 1 THEN R$ = "INSPIRAS"
380 IF R = 2 THEN R$ = "SOPLAS"
390 PRINT"ESTOY INSPIRANDO?"
400 PRINT"ESTOY SOPLANDO?"
410 INPUT B$
420 IF B$ = R$ THEN 460
430 GOTO 540

450 REM GANADOR
460 PRINT NOMBRE$
470 PRINT"HAS GANADO"
480 PRINT"NO PUEDO DERRIBAR TU CASA"
490 PRINT"SUYO FRUSTRADISIMO"
```

```
510 PRINT"LOBO"
520 GOTO 850
540 RETURN
550 REM SIGUE
560 PRINT"TU NUMERO DE JUGADAS HASTA
    AHORA ES"
570 PRINT JUGADAS
580 IF JUGADAS > 3 THEN 600
590 RETURN
600 REM INSPIRA-SOPLA
620 PRINT"(SHIFT) + (CLR HOME)"
630 PRINT"TU JUGADA NUMERO CUATRO"
640 PRINT"SIGNIFICA QUE TU CASA ES DE
    PAJA"
650 PRINT"HE INSPIRADO, HE SOPLADO Y
    HE DERRIBADO TU CASA"
660 PRINT"SUYO CONTENTISIMO"
670 PRINT"(SHIFT) + (CLR HOME)'
680 PRINT"LOBO"
700 GOTO 850
850 PRINT"ESCRIBE P PARA PARAR"
860 PRINT"ESCRIBE S PARA SEGUIR"
870 GET A$:IF A$ = ""THEN 870
880 IF A$ = "S"THEN GOTO 10
890 IF A$ = "P"THEN END
```

Si escribes y ejecutas el programa, observarás que la pantalla no queda muy clara. Es importante que la gente pueda leer fácilmente lo que hay en la pantalla. De manera que trabaja en el programa para hacer que quede un poco mejor. Utiliza todo lo que has aprendido.

Si te encuentras bloqueado en algo, pídele ayuda a alguien que sepa algo de programación. Hay muchas cosas que yo no comprendo y, a menudo, le pido ayuda a otras personas. Pero antes intento hacer las cosas por mí mismo. Por lo tanto, antes que nada, intenta mejorar el programa por tí mismo.



Jugar con un ordenador puede ser tan divertido como instructivo. La idea de este libro es ayudar a los más jóvenes y a sus padres para que comprendan cómo trabajar el Commodore 64 y como aprovechar al máximo sus posibilidades.
El libro está concebido de una manera sencilla y con un estilo simpático que demuestra que aprender puede ser divertido.
Una gran ventaja de la obra son los capítulos cortos, que invitan a niños y padres avanzar escalonadamente.
Una hora al día con el libro y el ordenador despertarán en el principiante una afición y unas ideas de programación que darán su fruto en el futuro.
Esta obra no es un libro de texto para aprender BASIC pero sí que es un sistema para conocer los comandos del BASIC del Commodore 64.
Niños a partir de siete años pueden entender este libro y por tanto con o sin sus padres podrán hacer uso del ordenador.
El autor, Meyer Solomon, es un afamado productor de Software cuya principal virtud es su interés por la enseñanza. Su afición a la informática le ha llevado a ser uno de los fundadores de la revista «Personal Computer World», con la que colabora habitualmente.

EDITORIAL NORAY, S.A.
San Gervasio de Cassolas, 79
08022 Barcelona